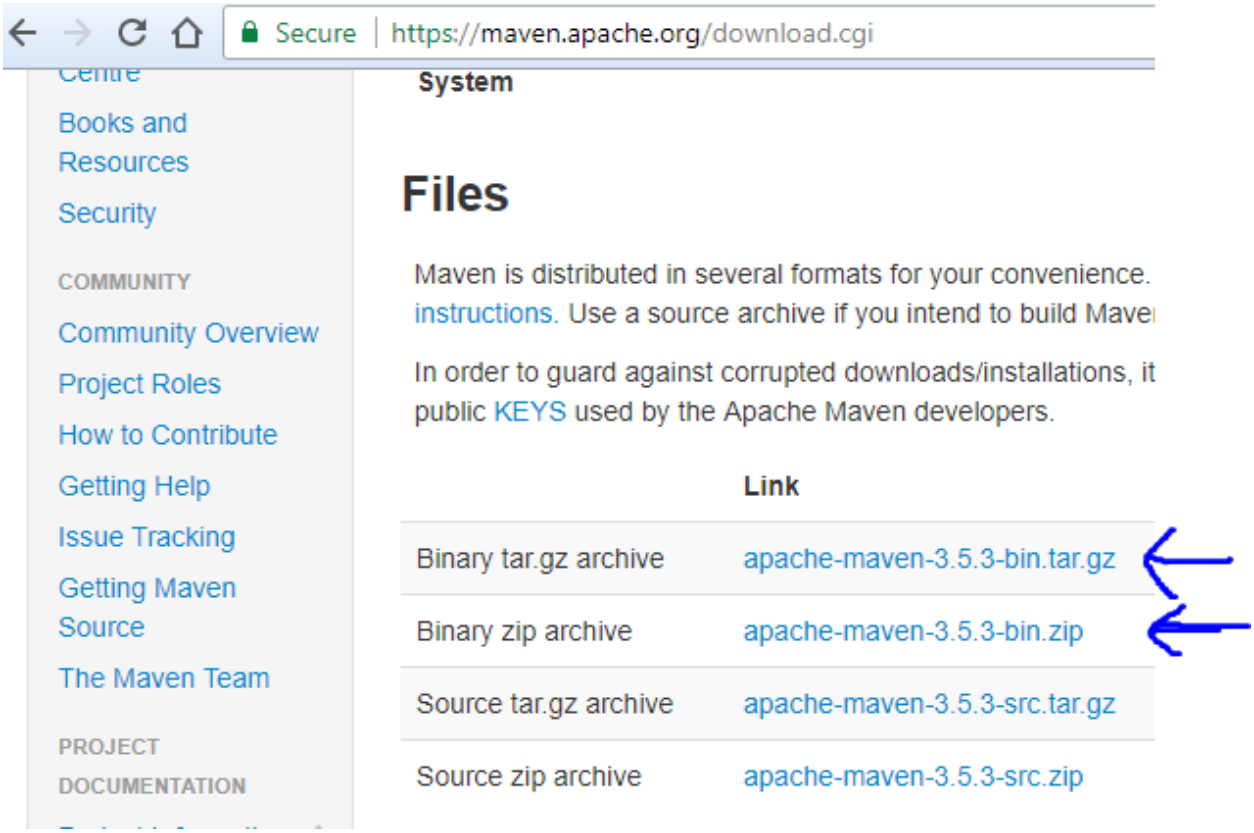
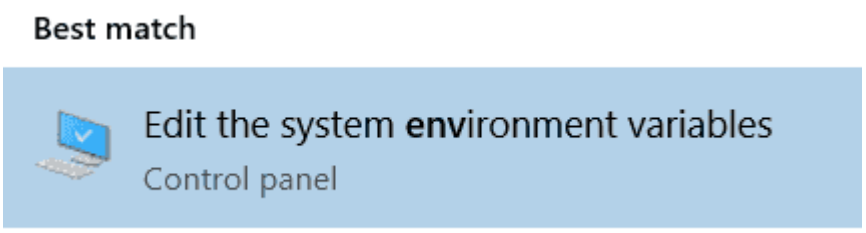
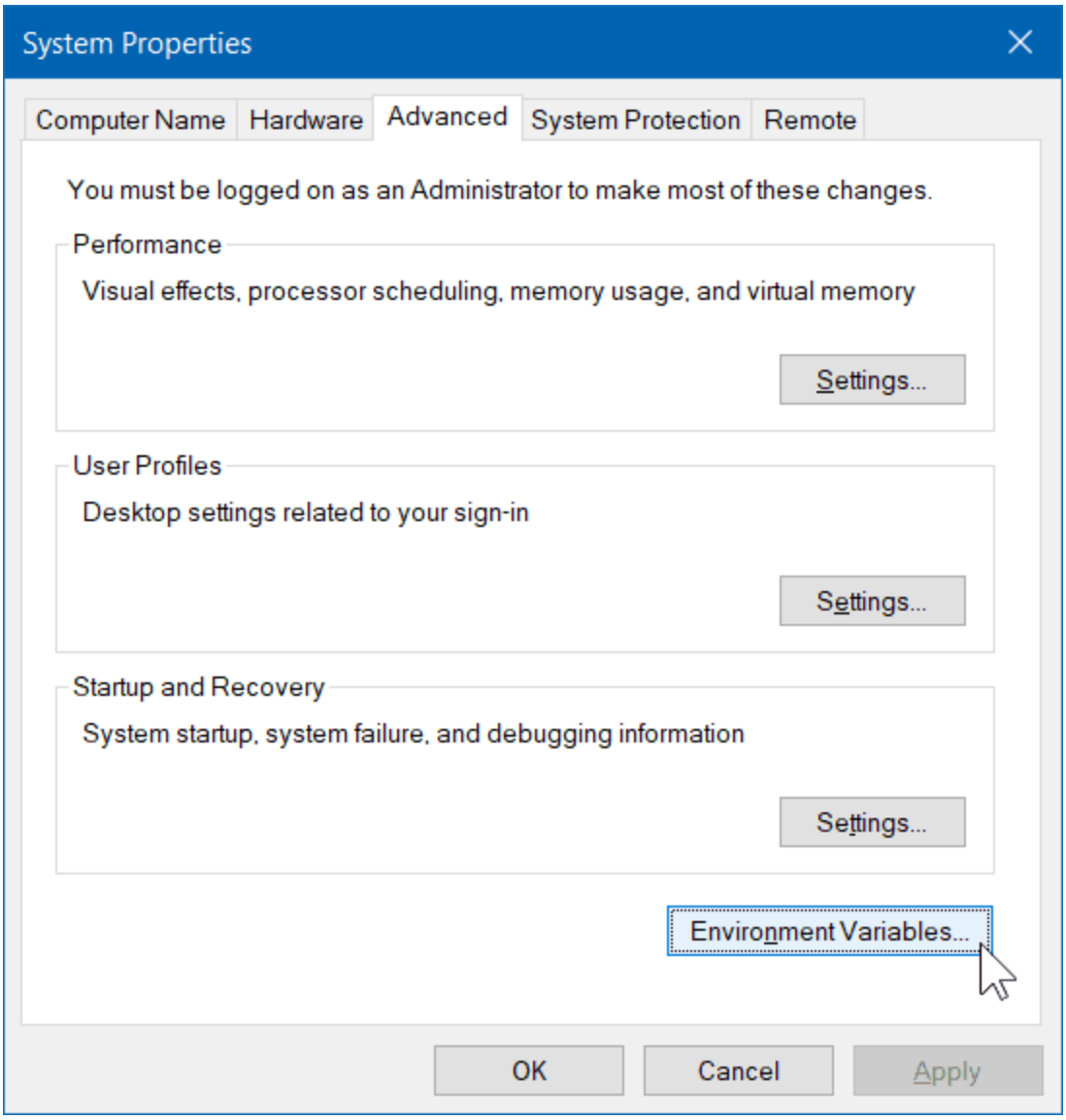
1. Intall WinPcap for windows (cmd: choco install winpcap)
2. Install maven : https://maven.apache.org/download.cgi



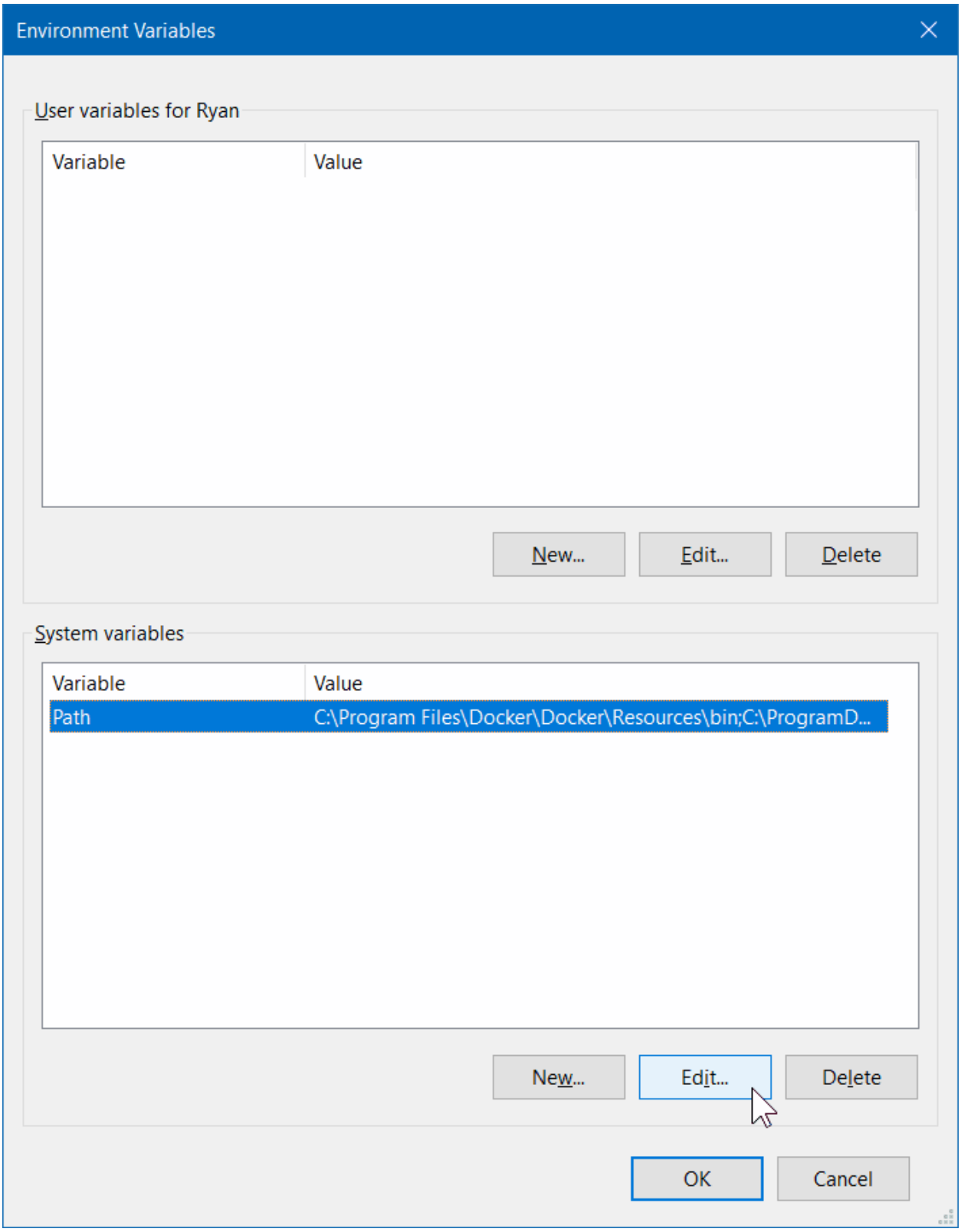
3. Unzip file and locate extracted apache-maven folder
4. Within open the /bin folder
5. Copy the file path
6. Open windows search and search for “env”, select “Edit the system environment varables”



7. Under Advanced tab, click “Environment Varables…”



8. Under “System Varables” locate Varable Path and click edit



9. Click “New” and paste copied path from step 4
10. Click “OK”
11. Test if maven is set up properly by running the command “mvn –version” in cmd

12. To generate a skeleton project that is ready to go… (NOTE: this will create a pcap directory)
    Enter into cmd: mvn archetype:generate -DgroupId=com.github.username -DartifactId=pcap -Dversion=1.0.0 -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
13. In cmd : “cd pcap” to enter the pcap directory
14. Open the “pom.xml” file for editing
15. Within the <dependencies></dependencies> section add:

```
<dependency>
  <groupId>org.pcap4j</groupId>
  <artifactId>pcap4j-core</artifactId>
  <version>1.8.2</version>
</dependency>
<dependency>
  <groupId>org.pcap4j</groupId>
  <artifactId>pcap4j-packetfactory-static</artifactId>
  <version>1.8.2</version>
</dependency>
```

16. Save and close
17. Now in the pcap folder, open the main folder
18. Open “java/com/github/username/” until you see the file App.java
19. Replace the contents of App.java with:

```
// App.java
package com.github.username;

import java.io.IOException;
import org.pcap4j.core.PcapNetworkInterface;
import org.pcap4j.util.NifSelector;

public class App
{
  public static void main(String[] args)
  {
    // The class that will store the network device
    // we want to use for capturing.
    PcapNetworkInterface device = null;

    // Pcap4j comes with a convenient method for listing
    // and choosing a network interface from the terminal
    try {
```

```
            // List the network devices available with a prompt
            device = new NifSelector().selectNetworkInterface();
        } catch (IOException e) {
            e.printStackTrace();
        }

        System.out.println("You chose: " + device);
    }
}
```

20. Go back to the folder with the pom.xml file and add in the <project></project> section:

```
<build>
    <plugins>
        <!-- Specify to the compiler we want Java 1.8 -->
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.7.0</version>
            <configuration>
                <source>1.8</source>
                <target>1.8</target>
            </configuration>
        </plugin>

        <!-- Tell the JAR plugin which class is the main class -->
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-jar-plugin</artifactId>
            <version>3.0.2</version>
            <configuration>
                <archive>
                    <manifest>
                        <mainClass>com.github.username.App</mainClass>
                    </manifest>
                </archive>
            </configuration>
        </plugin>

        <!-- Embed dependencies inside the final JAR -->
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-shade-plugin</artifactId>
            <version>3.1.0</version>
            <executions>
                <execution>
```

```
                    <phase>package</phase>
                    <goals>
                        <goal>shade</goal>
                    </goals>
                </execution>
            </executions>
            <configuration>
                <finalName>uber-${project.artifactId}-${project.version}</finalName>
            </configuration>
        </plugin>
    </plugins>
  </build>
```

21. Save and open cmd in the folder of pom.xml
22. To compile/run use
    "mvn package"
23. To launch App.java file:
    "java -jar target/uber-pcap-1.0.0.jar"
24. You MAY need to upgrade your privileges with sudo so use:
    "sudo java -jar target/uber-pcap-1.0.0.jar"

    It should run… Everything was based off the main website and this tutorial:
    https://www.devdungeon.com/content/packet-capturing-java-pcap4j